# DocLens: Intelligent Document Data Extraction and Verification System

1st Yogesh Parag Kulkarni
Computer *Science and Engg. Dept.*
Walchand Institute of Technology
*Solapur, Maharashtra, India*
yogeshpk0123@gmail.com

2nd Mr. A. R. Chinchawade
Assistant Professor
Computer *Science and Engg. Dept.*
Walchand Institute of Technology
*Solapur, Maharashtra, India*
archinchawade@witsolapur.org

3rd Archita Nilesh Yadav
Computer *Science and Engg. Dept.*
Walchand Institute of Technology
*Solapur, Maharashtra, India*
architanileshyadav@gmail.com

4th Irfan Jakir Hossain Shaikh
Computer *Science and Engg. Dept.*
Walchand Institute of Technology
*Solapur, Maharashtra, India*
shaikh.irfan.oct16@gmail.com

5th Abdul Basit Md. Idris Peerampalli
Computer *Science and Engg. Dept.*
Walchand Institute of Technology
*Solapur, Maharashtra, India*
apeerampalli@gmail.com

6th Gaurav Vilas Kulkarni
Computer *Science and Engg. Dept.*
Walchand Institute of Technology
*Solapur, Maharashtra, India*
gauravvkwitcse@gmail.com

7th Nilkil Arun Jain
Computer *Science and Engg. Dept.*
Walchand Institute of Technology
*Solapur, Maharashtra, India*
nikhilkumar1514@gmail.com

***Abstract*—**Millions of applications are submitted every year to various colleges, universities, and job opportunities. Despite the diversity among institutions and roles, the application process remains remarkably consistent across the board. Applicants must provide their personal details in text format and upload supporting documents as part of their submission. Supporting documents are typically submitted in image-based formats like JPEG, PDF, or PNG, which prevents direct, automated data processing. These documents are, therefore, manually verified against the details provided by the applicants. Recruiters and admission officers are burdened with the task of managing thousands of such applications, each accompanied by documents of a variety of formats, languages, and varying quality. Consequently, verification relies on manual review, an approach that is inherently slow, prone to inconsistency, and unable to scale, creating significant bottlenecks and risking decision-making accuracy.

To tackle this problem, we are developing an advanced, reliable document verification system designed to automate and streamline the verification process. Unlike traditional methods that rely heavily on Optical Character Recognition (OCR) and Natural Language Processing (NLP), our solution leverages state-of-the-art vision models with document-specific prompts. These models are capable of directly interpreting and extracting information from document images, regardless of format or language, with greater accuracy and contextual understanding. By utilizing vision models, our system can efficiently handle complex layouts, diverse fonts, and multi-language content, making the verification process more robust and reliable.

By automating the document verification process through vision models, Our solution directly addresses these challenges by automating the verification workflow, which demonstrably increases processing throughput and minimizes the risk of manual oversight. This innovative approach transforms recruitment and assessment workflows by reducing human error, improving efficiency, and enabling faster, more accurate decision-making.

***Keywords*—***Intelligent Document Processing (IDP), Document Verification, Data Extraction, Vision Models, Large Language Models (LLMs), Optical Character Recognition (OCR), Automated Verification, Real-time Processing, Multilingual Document Processing.*

## I. Introduction

In today's data-driven governance and enterprise operations, the volume of documents requiring verification—ranging from government-issued certificates to identity proofs—has increased exponentially. Legacy verification workflows, which depend on human inspection, are struggling to keep pace with the increasing volume of digital documents, presenting clear limitations in terms of speed, cost, and reliability. Requirement in the **recruitment processes**, **election administration**, **healthcare record verification**, **banking KYC compliance**, or validating documents like **land ownership** and **caste certificates** highlights a critical need across multiple sectors for a verification system that is not only automated but also intelligent and secure.

To address these multi-domain challenges, we propose a highly adaptable [22] **Intelligent Document Processing (IDP)** system that automates the extraction, understanding, and validation of information from semi-categorized or loosely structured documents. We engineered the proposed system to meet key operational targets, including near-instantaneous processing, validated accuracy, and a significant reduction in the need for manual review and trust in public and private services alike.

Our system moves beyond [2, 3, 8] traditional rigid template-matching or static OCR-based systems by leveraging [1, 4, 6] deep learning-based vision models capable of dynamic layout recognition, field-specific extraction, and multilingual processing. Unlike older systems that require precise formatting, our approach is robust against varied document formats and layouts, significantly enhancing scalability and reliability.

The core of our system leverages [15] **Meta's LLaMA-3.2 90B Vision-Instruct model**, a powerful open-source multimodal foundation model capable of processing both visual and textual data within documents. [16 - 18] Its open-source nature ensures **transparency, security, and extensibility**—key prerequisites for integration into critical infrastructures like government databases and financial institutions. Importantly, our system is **model-agnostic**, meaning that more advanced or fine-tuned models can be integrated with ease as technology evolves.

Our system is built not only to **automate verification tasks** but also to **support multilingual document processing**, achieve [21] **three-sigma accuracy**, and provide **real-time insights**—all while scaling across diverse use cases and geographical contexts. By combining the latest in

AI vision models, web technologies, and secure backend infrastructure, this IDP solution offers a **versatile framework** capable of revolutionizing how institutions across sectors validate documents and ensure compliance.

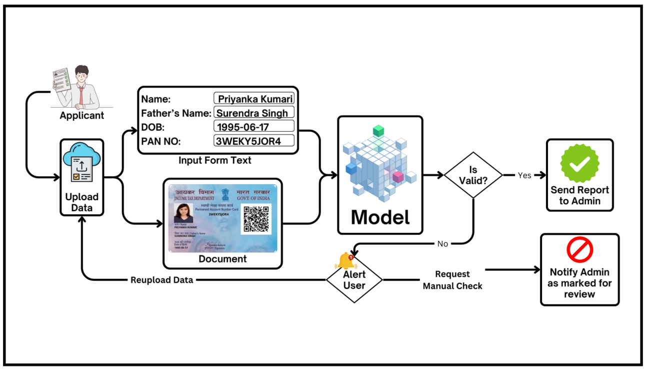


**Figure 1.1 : High Level Overview of System**

*This figure illustrates the overall architecture, showing interaction between the user, frontend, backend, database, and the vision model service.*

## II. Literature Review

In their work, Sinthuja et al. [1] approached text extraction by employing a hybrid deep learning architecture. They combined a Convolutional Neural Network (CNN) for visual feature extraction with a Bidirectional Long Short-Term Memory (BiLSTM) network to interpret the sequential structure of the text, reporting improved accuracy on both printed and handwritten content.

Similarly, Jiju et al. [2] present a two-stage approach for OCR on noisy images. Their method first utilizes the OpenCV library for image preprocessing, employing edge and contour detection to isolate the relevant document area. Subsequently, the cleaned image is processed by the Tesseract OCR engine with adaptive classification to enhance final recognition.

Addressing the challenge of unstructured medical data, Malashin et al. [3] developed a multi-stage pipeline. Their workflow begins with image preprocessing and OCR using tools like PyTesseract, followed by the application of specialized Natural Language Processing (NLP) libraries such as PullEnti and Natasha to perform named entity recognition on the extracted text.

The work by Misra et al. [4] centers on text detection through image feature analysis. Their method converts images to the HSV color space to identify potential Regions of Interest (ROIs). It then distinguishes text from non-text areas by analyzing a set of extracted features from these ROIs, including aspect ratio, pixel density, and edge information.

A significant contribution from Hansen et al. [5] is the creation of a large-scale, annotated corpus of PDF documents sourced from open-access repositories. On this dataset of over 31,000 pages, they trained and evaluated two deep learning models, one based on VGG-16 and another on Faster R-CNN, for semantic segmentation and object detection of document elements.

Ramakrishnan et al. [6] developed LA-PDFText, a tool specifically designed for layout-aware extraction from scientific PDFs. Its core mechanism involves merging word blocks based on spatial proximity and font features, followed by a rule-based classification using DROOLS to identify rhetorical categories and reconstruct the correct reading order.

Focusing on biomedical literature, Bui et al. [7] developed a system for classifying text in Cochrane reviews. Their approach utilizes a rule-based, multi-pass sieve algorithm to extract specific outcomes and evaluates its performance against standard machine learning alternatives, highlighting the effectiveness of domain-specific rules.

The challenges inherent in invoice data extraction are surveyed by Saout et al. [8]. They argue that success hinges on accurately modeling the document's key components and note that combining OCR with machine learning is a common and effective strategy for converting scanned invoices into structured, searchable data.

Raju et al. [9] tackle the problem of text extraction from video frames. Their methodology involves computing a set of thirteen features from frame blocks using various transformations (e.g., grayscale, DCT, FFT). These features are then used to train machine learning classifiers like Random Forest and J48 to categorize text within the video content.

Chao et al. [10] propose a structural decomposition approach for PDF analysis. Their method flattens a document into distinct text, image, and vector graphic layers. These layers are processed independently before being programmatically reassembled into a structured XML output, preserving the document's original layout and content.

An unsupervised approach to document understanding is presented by Klampfl et al. [11]. Instead of relying on pre-trained models, their pipeline uses geometric relations and font information, employing clustering techniques to distinguish body text from the table of contents in scientific articles, outperforming some supervised methods.

Finally, Luong et al. [12] developed SectLabel, a module that employs Conditional Random Fields (CRFs) to identify the logical structure of scholarly documents. The key to their method's improved performance is the integration of rich features derived from OCR, such as font size and text position, into the CRF model.

**Identified Gaps and DocLens Advantages Summary:**

Our review of existing literature [1-12] reveals several common limitations. Many approaches, for instance, are constrained by rigid, template-based designs ([6, 7, 11]) or rely on a sequential, and therefore error-prone, process of OCR followed by a separate NLP analysis ([2, 3, 8]). DocLens distinguishes itself by using a unified vision-language model to bypass these issues. This approach provides an end-to-end solution integrating highly accurate extraction with immediate verification, offering superior adaptability, multilingual capabilities, and efficiency across various sectors.

## III. Technology Stack

1. **Core Intelligence**: **Vision-Language Model:** At the heart of our system is the meta-llama/Llama-3.2-90B-Vision-Instruct model. We selected this specific vision-language model for its state-of-the-art multimodal capabilities, which allow it to directly interpret both the visual layout and textual content of a document in a single pass. Its open-source nature was a critical factor, ensuring transparency and allowing for future fine-tuning on domain-specific datasets. This model's ability to process visual prompts makes it superior to traditional OCR/NLP pipelines for handling diverse document layouts.

2. **Backend Framework: Django:** For the server-side logic, we chose Django, a high-level Python web framework. Its "batteries-included" philosophy provided a robust and secure foundation for developing our API endpoints and managing the verification workflow. The built-in Object-Relational Mapper (ORM) was particularly advantageous, as it streamlined all database interactions with PostgreSQL and allowed us to define our data models in a clean, maintainable way.

3. **Database Management: PostgreSQL:** Our data persistence layer is handled by PostgreSQL. This relational database was chosen over other options for two primary reasons: its proven reliability for complex queries and its native support for vector data types through the pgvector extension. This extension is essential for our system, as it allows for the efficient storage and indexed retrieval of the spatial vectors and visual embeddings that define each document template's layout.

4. **Frontend Interface: React.js:** To deliver a highly responsive and intuitive user experience, the frontend was developed using the React.js library. Its component-based architecture enabled the rapid development of a modular interface, including dynamic forms for applicant data entry, real-time feedback on verification status, and interactive dashboards for administrators. This choice ensures a modern, seamless interaction with the system's powerful backend capabilities.

## IV. DOCLENS

The developed system is divided into two key phases, which are mentioned below:

**Phase 1: Document Template Configuration (One-Time Training & Setup Phase)**

1. **Analyzing Sample Documents for Prompt Context:** The setup phase begins by assembling a representative set of sample documents (e.g., various Aadhaar card versions, PAN cards). These samples serve as the basis for understanding the visual structure and variations of each document type. Vision models analyze these samples to identify key visual landmarks and structural elements like headers, logos, photograph locations, and specific layout patterns. These identified features become crucial contextual anchors for the prompts.
2. **Defining the Prompt's Core Instructions (Fields & Queries):** For each document type, administrators explicitly define the target information fields (e.g., FullName, DateOfBirth, DocumentID). This definition forms the core of the prompt, specifying *what* the AI needs to find. Each field is associated with detailed instructions within the prompt, including:
   - **Expected Data Type:** Text, Date, Numeric String, etc.
   - **Format Constraints:** Regex patterns (e.g., \d{12}) or specific date formats (DD/MM/YYYY).
   - **Keywords:** Text labels often found near the target field (e.g., "Name:", "Date of Birth / जन्म तिथि").
   - **Relative Location Hints:** Initial guidance on where the field typically resides in relation to the structural anchors identified earlier (e.g., "below header," "right of photo").
3. **Learning and Encoding Spatial Instructions:** Advanced vision models further analyze the samples, guided by the initial prompt definitions and structural anchors. They learn the precise probable locations (bounding boxes) of the target text fields relative to these anchors. This spatial understanding – where each piece of information is typically located within the specific document layout – is encoded as a critical part of the prompt. This allows the AI to efficiently pinpoint where to apply Optical Character Recognition (OCR) during extraction, rather than scanning the entire document.
4. **Storing the Configured Prompts:** The culmination of this phase is the storage of a complete prompt configuration package for each document type in a secure database. This package encapsulates:
   - The defined field specifications and associated metadata (data types, formats, keywords).
   - The learned spatial instructions (typical relative coordinates/bounding boxes for each field).
   - Compact visual embeddings that represent the overall expected layout, enabling rapid document type identification later.

**Phase 2: Application Verification Process**

1. Applicant Data Entry and Document Upload: The applicant interacts with a digital form, inputting their personal details (Name, DoB, Aadhaar Number, etc.). They are then guided to upload digital copies (scans or photos) of the necessary supporting documents into designated fields, each field implicitly or explicitly linked to a specific document type (e.g., "Upload Aadhaar," "Upload PAN Card").
2. **Input Field Pre-processing and Validation:** Before initiating document analysis, the system performs preliminary checks on the applicant's manually entered data. This includes basic format validation (phone, email, date), checksum calculations for identifiers like Aadhaar numbers ([22] Verhoeff algorithm) to detect obvious typos, and ensuring all required fields are filled.
3. **Applying Pre-Associated Template for Targeted Extraction:** Based on the specific input field used for the upload, the system **retrieves the corresponding pre-configured document template** (established in Phase 1) from the database. This template contains the precise instructions for the expected document type. The system then executes the instructions within this template:
   A. It utilizes the template's stored spatial information (field locations relative to known anchors like logos, photos) to direct the vision models.
   B. The vision models focus their analysis *only* on the specific regions identified in the template for the required data fields.
   C. Optical Character Recognition (OCR) is applied precisely within these targeted bounding boxes, as dictated by the template, to extract the relevant text efficiently by the vision model.
4. **Processing Extracted Text (OCR Output):** The raw text data resulting from the targeted OCR undergoes essential cleaning and standardization. This involves removing noise artifacts, normalizing text case, trimming whitespace, converting diverse date formats into a consistent canonical format (e.g., YYYY-MM-DD), and applying rules or algorithms to handle variations in name order (e.g., standardizing "Doe Jane" vs. "Jane Doe"). This normalized text is prepared for accurate comparison.
5. **Automated Comparison:** The system compares the cleaned information extracted from the document against the information the applicant entered in the form. It checks if the values match according to set rules:

A. **Exact Match:** Critical details like ID numbers must match perfectly.

B. **Direct Match:** Standardized dates and numbers are compared directly for equality.

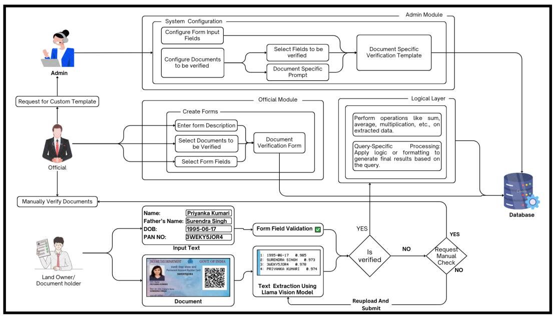
C. **Flexible Match** (Fuzzy Matching): Fields like names and addresses can have minor differences but still be considered a match if they are very similar. Example:

- Applicant entered Name: "Robert Downey Jr"
- Document extracted Name: "Robert Downy Jr." (slight typo in 'Downey', period added)

To account for such minor variations, the system calculates a similarity score using a fuzzy matching algorithm, specifically Levenshtein distance [13].

6. **Successful Verification Path:** If the comparison confirms that the data extracted via the template matches the applicant's entered data within the acceptable thresholds, the verification is successful. A comprehensive verification report is generated, detailing the inputs, the extracted outputs, the match status for each field, confidence scores (if applicable), and a verification timestamp. The application's status is updated accordingly (e.g., "Verified").
7. **Mismatch Handling:** If the comparison reveals significant discrepancies between the template's extracted output and the applicant's entered data (falling outside acceptable thresholds), the system flags the application.

The applicant might be notified about the specific mismatch. They are usually given options: either correct their form input and potentially re-upload the document (triggering a re-application of the same template) or request a manual review. If manual review is chosen, the complete application details —including entered data, documents, the template's extracted output, and highlighted discrepancies—are forwarded to a human operator queue for final assessment.

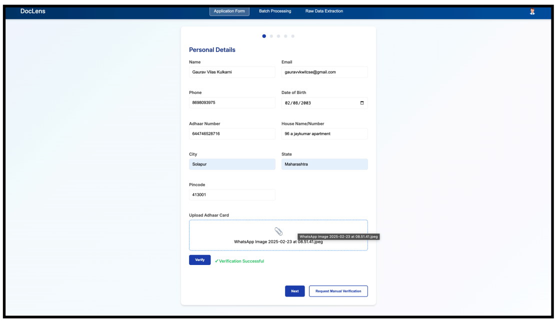

Admin Module
System Configuration
Configure Form Input Fields
Configure Documents to be verified
Select Fields to be verified
Document Specific Prompt
Document Specific Verification Template
Admin
Request for Custom Template
Official Module
Create Forms
Enter form Description
Select Documents to be Verified
Select Form Fields
Document Verification Form
Logical Layer
Perform operations like sum, average, multiplication, etc., on extracted data.
Query-Specific Processing: Apply logic or formatting to generate final results based on the query.
Official
Database
Manually Verify Documents
Input Text
Form Field Validation
YES
Is verified
NO
YES
Request Manual Check
NO
Land Owner/ Document holder
Document
Text Extraction Using Llama Vision Model
Reupload And Submit

Name: Priyanka Kumari
Father's Name: Surendra Singh
DOB: 1995-06-17
PAN NO: 3WEKY5JOR4

```
1: 1995-06-17   0.985
2: SURENDRA SINGH   0.973
3: 3WEKY5JOR4   0.978
4: PRIYANKA KUMARI   0.974
```

**Figure 4.1 : Flow Diagram**

*This diagram outlines the step-by-step process flow for both the template configuration phase and the application verification phase.*



Personal Details

Name: Gaurav Vilas Kulkarni

Email: gauravvkwitcse@gmail.com

Phone: 8698093975

Date of Birth: 02/08/2003

Adhaar Number: 644746528716

House Name/Number: 96 a jaykumar apartment

City: Solapur

State: Maharashtra

Pincode: 413001

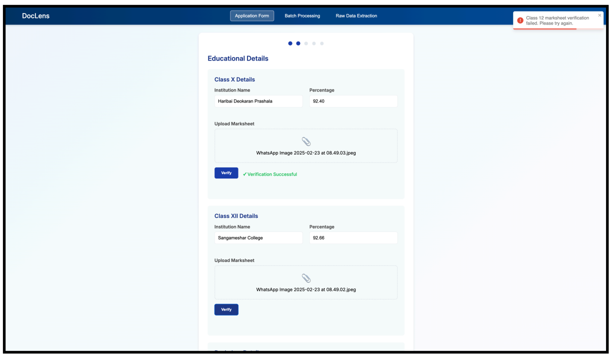
Upload Adhaar Card: WhatsApp Image 2025-02-23 at 08.51.41.jpeg

Verify ✓Verification Successful

Next | Request Manual Verification

**Figure 4.2: Application Form (Personal Details Verification Successful) UI**

This UI shows the applicant entering personal details (like name, DoB) and uploading their Aadhaar card, initiating the Phase 2 automated verification process and successful verification



Class 12 marksheet verification failed. Please try again.

Educational Details

Class X Details

Institution Name: Haribai Deokaran Prashala

Percentage: 92.40

Upload Marksheet: WhatsApp Image 2025-02-23 at 08.49.03.jpeg

Verify ✓Verification Successful

Class XII Details

Institution Name: Sangameshar College

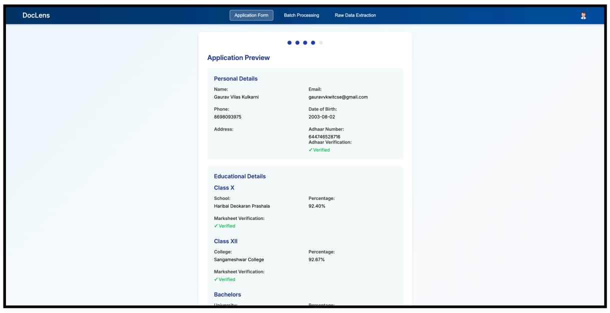
Percentage: 92.66

Upload Marksheet: WhatsApp Image 2025-02-23 at 08.49.02.jpeg

Verify

**Figure 4.3: Educational Details (Verification Unsuccessful Case) UI**

*This UI shows an applicant's entered marks/percentage for recruitment, marked as unsuccessful verification due to a mismatch with data extracted from the uploaded marksheet.*



Application Preview

Personal Details

Name: Gaurav Vilas Kulkarni

Email: gauravvkwitcse@gmail.com

Phone: 8698093975

Date of Birth: 2003-08-02

Address:

Adhaar Number: 644746528716

Adhaar Verification: ✓Verified

Educational Details

Class X

School: Haribai Deokaran Prashala

Percentage: 92.40%

Marksheet Verification: ✓Verified

Class XII

College: Sangameshwar College

Percentage: 92.67%

Marksheet Verification: ✓Verified

Bachelors

**Figure 4.4: Application Preview UI**

*This UI summarizes verification results for all submitted details, showing pass/fail status. Submission requires full verification, but offers a manual review request for failed items.*

## V. Use Cases of DocLens Across Multiple Sectors

| Sector | Document Types | Purpose |
|---|---|---|
| **Recruitment Process** | ID proofs, educational certificates, experience letters, resumes | Automate candidate verification and reduce hiring time |
| **Admission in Institutions** | ID cards, marksheets, transfer certificates, category certificates | Fast-track admission document verification and ensure eligibility |
| **Land and Property Record Management** | Sale deeds, 7/12 extracts, property cards, building plans, property layouts | Accelerate land transactions and prevent fraudulent sales |
| **Healthcare Sector** | Medical records, lab reports, insurance documents | Speed up patient data processing and insurance claims |
| **Banking and Financial Sector** | KYC documents, bank statements, property papers, ITRs, salary slips | Automate customer onboarding and loan processing |

| | | |
|---|---|---|
| **Insurance Sector** | Policyholder IDs, claim documents, FIRs, medical bills | Enable faster insurance policy issuance and claim settlements |

**Table 5.1 : Sector-wise Application of DocLens**

## VI. Results and Analysis

Under ideal conditions with clear, well-scanned documents, the DocLens system demonstrates extraction accuracy approaching 100%.

System performance degrades predictably with deteriorating image quality; significant blurring or low-resolution scans were the primary cause of reduced accuracy, as shown in Table 6.1.

To demonstrate the system's performance, we evaluated DocLens over a dataset of 1000 documents across different sectors (Recruitment, Admission, Voting, Land Records, Banking, Healthcare).

| Condition | Accuracy (%) |
|---|---|
| Clear and properly scanned documents | 100% |
| Slightly noisy or compressed documents | 99.5% |
| Blurred/low-quality images | 92% |

**Table 6.1 : Accuracy Analysis**

Observations:

- When documents are clearly scanned or captured, DocLens achieves perfect extraction and verification.
- Slight noise or compression has negligible effect on performance.
- Heavy blurring or unclear text lowers accuracy but remains above 90%.

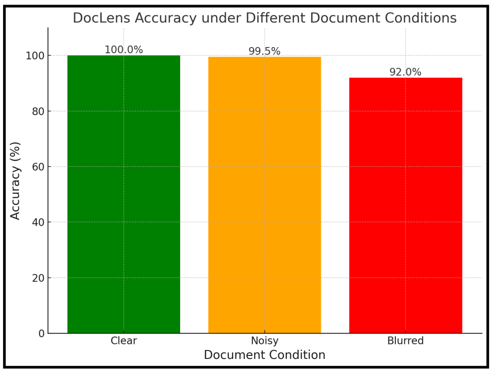


**Fig 6.1 : Graph - DocLens Accuracy**

## VII. Future Work

Based on our current results, we have identified several key areas for future development:

1. **Expanded Document Type Support:** Develop and test pre-configured prompts for a wider range of documents (e.g., passports, driving licenses, bank statements, complex legal contracts).
2. **Enhanced Low-Quality Image Handling:** Integrate advanced image pre-processing techniques (denoising, deblurring, super-resolution) specifically tailored for document images to further improve accuracy on very poor-quality inputs.
3. **Advanced Fraud Detection:** Incorporate capabilities to detect signs of digital tampering or forgery within documents (e.g., inconsistent fonts, signs of editing, metadata analysis).
4. **Model Fine-Tuning:** Fine-tune the underlying vision-language models on domain-specific datasets (e.g., healthcare records, financial documents) to potentially boost accuracy and handle specialized jargon or layouts more effectively.
5. **Integration with External Databases/APIs:** Enable direct verification against official government databases or other trusted third-party sources e.g. DigiLocker where APIs are available (e.g., verifying Aadhaar status directly).

## VIII. Conclusion

In this paper, we presented DocLens, an IDP system that effectively uses Vision Models to automate data extraction and verification from semi-structured documents.

The system demonstrates a near-**100% accuracy rate** in field-level data extraction and verification across diverse document formats, including government-issued IDs, academic certificates, land records, and healthcare forms. Furthermore, it maintains a processing time of **under three seconds per document**, provided stable internet connectivity is available, thereby satisfying the time-critical requirements of large-scale document-driven processes.

The system addresses the major challenges in current document processing pipelines by eliminating the need for:

- Region-specific field selection
- Manual rule-based verification
- Language-specific OCR engines
- Static template dependence.

In doing so, DocLens satisfies all of the defined objectives for modern IDP systems:

- Automated data or information extraction and verification
- Minimal manual intervention
- Support for multilingual documents
- High accuracy adhering to the Three Sigma standard
- Real-time user feedback
- High throughput suitable for production environments.

The flexibility of this model-driven approach ensures its applicability in diverse sectors, including recruitment, finance, and public services. With its high performance and adaptability, DocLens provides a viable framework for developing the next generation of automated and intelligent document verification solutions.

## Reference

[1] M Sinthuja, Chirag Ganesh Padubidri, Gaddam Sai Jayachandra, Mudduluru Charan Teja, Golthi Sai Pavan Kumar. Received on 23 Sept. 2015, Revised 22 March 2016, Accepted 31 March 2016, Published 1 Apr. 2016, "Extraction of Text from Images Using Deep Learning." Procedia Computer Science, 235, 789–798. https://doi.org/10.1016/j.procs.2024.04.075

[2] Alan Jiju, Shaun Tuscano, and Chetana Badgujar (2021). “OCR Text Extraction.” International Journal of Engineering and Management Research, Volume-11, Issue-2 (April 2021). https://doi.org/10.31033/ijemr.11.2.11

[3] Ivan Malashin, Igor Masich, Vadim Tynchenko, Andrei Gantimurov, Vladimir Nelyub, and Aleksei Borodulin. Received 21 Apr. 2024, Revised 12 June 2024, Accepted 16 June 2024, Published 18 June 2024 “Image Text Extraction and Natural Language Processing of Unstructured Data from Medical Reports.” Machine Learning and Knowledge Extraction, 6, 1361–1377. https://doi.org/10.3390/make6020064

[4] Chinmaya Misra, P K Swain, Jibendu KUMAR Mantri (2012). “Text Extraction and Recognition from Image using Neural Network.” International Journal of Computer Applications, Volume 40, No. 2. DOI: 10.5120/4927-7156

[5] Matthias Hansen, André Pomp, Kemal Erki and Tobias Meisen (2019). “Data-Driven Recognition and Extraction of PDF Document Elements.” Technologies, 7, 65. doi:10.3390/technologies7030065

[6] Cartic Ramakrishnan, Abhishek Patnia, Eduard Hovy, and Gully APC Burns (2012). "Layout-aware text extraction from full-text PDF of scientific articles." Source Code for Biology and Medicine, 7:7. http://www.scfbm.org/content/7/1/7

[7] Duy Duc An Bui, Guilherme Del Fiol, Siddhartha Jonnalagadda (2016). "PDF text classification to leverage information extraction from publication reports." Journal of Biomedical Informatics, 61, 141–148. http://dx.doi.org/10.1016/j.jbi.2016.03.026

[8] Thomas Scout, Frederic Lardeux, and Frederic Saubion. Received 10 Jan. 2024, Accepted 27 Jan. 2024, Published 31 Jan. 2024, Current Version 9 February 2024 “An Overview of Data Extraction From Invoices." IEEE Access, 12. DOI: 10.1109/ACCESS.2024.3360528

[9] Nidhin Raju, Dr. Anita H.B (2017). “Text Extraction from Video Images.” International Journal of Applied Engineering Research, ISSN 0973-4562 Volume 12, Number 24, pp. 14750-14754. © Research India Publications. http://www.ripublication.com

[10] Hio Chao, Jian Fan (2004). "Layout and Content Extraction for PDF Documents." In: Document Analysis Systems VI. DAS 2004. Lecture Notes in Computer Science, vol 3163. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-28640-0_20

[11] S. Klampfl, R. Kern, “An unsupervised machine learning approach to body text and table of contents extraction from digital scientific articles”, Research and Advanced Technology for Digital Libraries, Springer, 2013, pp. 144–155.

[12] M.T. Luong, T.D. Nguyen, M.Y. Kan, Logical structure recovery in scholarly articles with rich document features, Multimedia Storage Retrieval Innovat.Digital Lib. Syst. (2012) 270.

[13] Bonnie Berger, Michael S. Waterman, and Yun William Yu (2021). “Levenshtein Distance, Sequence Comparison and Biological Database Search.” *IEEE Transactions on Information Theory*, VOL. 67, NO. 6, pp. 3287-3316. DOI: 10.1109/TIT.2021.3073724

[14] React.js. https://react.dev/

[15] Meta Llama. https://huggingface.co/meta-llama/Llama-3.2-90B-Vision-Instruct

[16] Heron, Michael; Hanson, Vicki; Ricketts, Ian (2013). “Open Source and Accessibility: Advantages and Limitations.” Journal of Interaction Science, 1, 2. DOI: 10.1186/2194-0827-1-2.

[17] Almarzouq, Mohammad; Zheng, Li; Rong, Guang; Grover, Varun (2005). “Open Source: Concepts, Benefits, and Challenges.” *Communications of the Association for Information Systems*, 16, pp. 756-784. DOI: 10.17705/1CAIS.01637.

[18] Ahmad, N.; Tripathi, N. (2024). [Software Business. ICSOB 2023] – “Benefits, Challenges, and Implications of Open-Source Software for Health-Tech Startups: An Empirical Study.” In: *Lecture Notes in Business Information Processing*, vol 500. Springer, Cham. https://doi.org/10.1007/978-3-031-53227-6_19

[19] Django. https://docs.djangoproject.com/en/5.2/

[20] PostgreSQL - PGVector : Vector storing in PostgreSQL (2024). "pgvector 0.7.0 released." PostgreSQL News. https://www.postgresql.org/about/news/pgvector-070-released-2852/

[21] Indeed (n.d.). "3 Sigma. (Standard Deviation) Statistical Measure" Indeed Career Advice. https://www.indeed.com/career-advice/career-development/3-sigma

[22] Sharath, Krs (n.d.). "How Aadhar Number is generated and Validated - Verheoff Algorithm." Medium. https://medium.com/@krs.sharath03/how-aadhar-number-is-generated-and-validated-3c3e7172e606

[23] Esposito, Floriana; Ferilli, Stefano; Basile, T.M.A.; Di Mauro, Nicola (2005). [International Conference on Document Analysis and Recognition (ICDAR) 2005] – “Intelligent document processing.” Proceedings of the International Conference on Document Analysis and Recognition, ICDAR. Vol. 2, pp. 1100 - 1104. DOI: 10.1109/ICDAR.2005.144.